Related-Key Rectangle Attack on the Full SHACAL-1

Orr Dunkelman*[1], Nathan Keller**[2], Jongsung Kim***[3,4]

[1]Computer Science Department, Technion.
Haifa 32000, Israel
orrd@cs.technion.ac.il
[2]Einstein Institute of Mathematics, Hebrew University.
Jerusalem 91904, Israel
nkeller@math.huji.ac.il
[3] ESAT/SCD-COSIC, Katholieke Universiteit Leuven
Kasteelpark Arenberg 10, B-3001 Leuven-Heverlee, Belgium
Kim.Jongsung@esat.kuleuven.be
[4] Center for Information Security Technologies(CIST), Korea University
Anam Dong, Sungbuk Gu, Seoul, Korea
joshep@cist.korea.ac.kr

Abstract. SHACAL-1 is a 160-bit block cipher with variable key length of up to 512-bit key based on the hash function SHA-1. It was submitted to the NESSIE project and was accepted as a finalist for the 2nd phase of the evaluation.

In this paper we devise the first known attack on the full 80-round SHACAL-1 faster than exhaustive key search. The related-key differentials used in the attack are based on transformation of the collision-producing differentials of SHA-1 presented by Wang et al.

1 Introduction

In 1993, NIST has issued a standard hash function called Secure Hash Algorithm (FIPS-180) [25]. Later this version was named SHA-0, as NIST published a small tweak to this standard called SHA-1 in 1995. Both SHA-0 and SHA-1 are based on padding the message and dividing it to blocks of 512 bits, and then iteratively compressing those blocks into a 160-bit digest. Recently, NIST has published three more standard hash functions as part of FIPS-180: SHA-256, SHA-384

* The author was supported by the Clore scholarship programme and by the Israel MOD Research and Technology Unit..

** The author was supported by the Adams fellowship.

*** This author was financed by a Ph.D grant of the Katholieke Universiteit Leuven and by the Korea Research Foundation Grant funded by the Korean Government(MOEHRD) (KRF-2005-213-D00077) and supported by the Concerted Research Action (GOA) Ambiorics 2005/11 of the Flemish Government and by the European Commission through the IST Programme under Contract IST2002507932 ECRYPT.

and SHA-512. Each of the new hash functions has a digest size corresponding to its number, i.e., SHA-256 has a 256-bit digest, etc. After the publication of these hash functions, NIST has issued another hash function SHA-224 that has a digest size of 224 bits.

Both SHA-0 and SHA-1 were subjected to a great deal of analysis [11]. In the last two years there was a major progress in the attacks on both of the hash functions. This progress included finding a collision in SHA-0, and devising an algorithm that can find a collision in SHA-1 in less than 2^{63} SHA-1 applications [3, 4, 30, 32, 28]. The new techniques are based on finding good differentials of the compressing function of SHA-1 and combining them with some novel plaintext modification techniques.

It was suggested to use the compression function of SHA-1 as a block cipher [13]. Later this suggestion was named SHACAL-1 and submitted to the NESSIE project [14]. SHACAL-1 is a 160-bit block cipher with variable key length (0–512 bits) and 80 rounds based on the compression function of SHA-1. The cipher, was selected as a NESSIE finalist, but was not selected for the NESSIE portfolio [22].

Due to the structure of SHACAL-1, differentials of SHA-1 correspond to related-key differentials of SHACAL-1. Hence, it seems natural that some of the techniques used in the new attacks on SHA-1 can be converted into a related-key attack on SHACAL-1. We show that this is indeed the case. The differentials found in the attacks devised in [30] can be converted into high probability related-key differentials of SHACAL-1.

After transforming the collision producing differentials into related-key differentials, we use them in a related-key rectangle attack [8, 15, 18]. The resulting attack succeeds to attack the full 80-round SHACAL-1 using four related-keys faster than exhaustive key search.

The related-key rectangle technique was used in previously published attacks on SHACAL-1 [15, 18] and was by far the most successful technique to attack the cipher. The best previously known attack on the cipher based on this technique was applicable up to 70 rounds of SHACAL-1. Our results extend these previously known results by using improved differentials and improved attack techniques.

We note that the best known attack on SHACAL-1 that does not use related-keys is a rectangle attack on 49-round SHACAL-1 [7]. A comparison of the known attacks along with our new results on SHACAL-1 is presented in Table 1.

This paper is organized as follows: In Section 2 we describe the block cipher SHACAL-1. In Section 3 we describe the previously known results on SHACAL-1 and the relevant results on SHA-1. In Section 4 we give a short description of the related-key rectangle attack. In Section 5 we present the new attacks on the full SHACAL-1. Section 6 explores the differences between our attacks on SHACAL-1 and other works on SHA-1. The appendix contains the differentials used in the attack. Finally, Section 7 summarizes the paper.

Attack & Source	Number of Keys	Number of Rounds	Rounds	Complexity Data	Complexity Time
Differential [20]	1	41	0–40	2^{141} CP	2^{491}
Amplified Boomerang [20]	1	47	0–46	$2^{158.5}$ CP	$2^{508.4}$
Rectangle [7]	1	47	0–46	$2^{151.9}$ CP	$2^{482.6}$
Rectangle [7]	1	49	29–77	$2^{151.9}$ CC	$2^{508.5}$
Related-Key Rectangle [18]	2	59	0–58	$2^{149.7}$ RK-CP	$2^{498.3}$
Related-Key Rectangle [15]	4	70	0–69	$2^{151.8}$ RK-CP	$2^{500.1}$
Related-Key Rectangle (Section 5.2)	4	80	0–79	$2^{159.8}$ RK-CP	$2^{423.0}$
Related-Key Rectangle (Section 5.2)	4	80	0–79	$2^{153.8}$ RK-CP	$2^{504.2}$

Complexity is measured in encryption units.
CP — Chosen Plaintexts, CC — Chosen Ciphertexts, RK — Related-Key

Table 1. Summary of Our Results and Previously Known Results on SHACAL-1.

2 Description of SHACAL-1

SHACAL-1 [14] is a 160-bit block cipher supporting variable key lengths (0–512 bits). It is based on the compression function of the hash function SHA-1 [25]. The cipher has 80 rounds (also referred as steps) grouped into four types of 20 rounds each.[1]

The 160-bit plaintext is divided into five 32-bit words – A, B, C, D and E. We denote by X_i the value of word X before the ith round, i.e., the plaintext P is divided into A_0, B_0, C_0, D_0 and E_0, and the ciphertext is composed of $A_{80}, B_{80}, C_{80}, D_{80}$ and E_{80}.

In each round the words are updated according to the following rule:

$$\begin{aligned}
A_{i+1} &= W_i + ROTL_5(A_i) + f_i(B_i, C_i, D_i) + E_i + K_i \\
B_{i+1} &= A_i \\
C_{i+1} &= ROTL_{30}(B_i) \\
D_{i+1} &= C_i \\
E_{i+1} &= D_i
\end{aligned}$$

where + denotes addition modulo 2^{32}, $ROTL_j(X)$ represents rotation to the left by j bits, W_i is the round subkey, and K_i is the round constant.[2] There are three different functions f_i, selected according to the round number:

$$\begin{array}{lll}
f_i(X,Y,Z) = f_{if} = (X \& Y) | (\neg X \& Z) & & 0 \le i \le 19 \\
f_i(X,Y,Z) = f_{xor} = (X \oplus Y \oplus Z) & & 20 \le i \le 39, 60 \le i \le 79 \\
f_i(X,Y,Z) = f_{maj} = ((X \& Y) | (X \& Z) | (Y \& Z)) & & 40 \le i \le 59
\end{array}$$

[1] To avoid confusion, we adopt the common notations for rounds. In [14] the notation step stands for round, where round is used for a group of 20 steps.

[2] This time we adopt the notations of [14], and alert the reader of the somewhat confusing notations.

In [14] it is strongly advised to use keys of at least 128 bits, even though shorter keys are supported. The first step in the key schedule algorithm is to pad the supplied key into a 512-bit key. Then, the 512-bit key is expanded into eighty 32-bit subkeys (or a total of 2560 bits of subkey material). The expansion is done in a linear manner using a linear feedback shift register (over $GF(2^{32})$).

The key schedule is as follows: Let $M_0, \ldots, M_{15}$ be the 16 key words (32 bits each). Then the round subkeys $W_0, \ldots, W_{79}$ are computed by the following algorithm:

$$W_i = \begin{cases} M_i & 0 \leq i \leq 15 \\ (W_{i-3} \oplus W_{i-8} \oplus W_{i-14} \oplus W_{i-16}) \lll 1 & 16 \leq i \leq 79 \end{cases}$$

3 Previous Results

A preliminary differential and linear analysis of the properties of the compression function of SHA-1 as a block cipher is presented in [13]. The found differentials are relatively short (10 rounds) and have probabilities varying between 2^{-13} and 2^{-26} (depending on the round functions).

In [26] these differentials are improved, and 20-round differentials with probability 2^{-41} are presented. In [20] another set of differentials of SHACAL-1 is presented, including a 30-round differential with probability 2^{-130}.

In [24] an algorithm for identifying whether two SHACAL-1 encryptions use a pair of related keys is presented. The attack is based on finding slid pairs. Once a slid pair is encountered, the attacker can determine whether the two encryptions have related keys. The attack requires about 2^{96} encryptions under each of the two keys to find a slid pair.

In [20] a 21-round differential for rounds 0–20 and a 15-round differential for rounds 21–35 are combined to devise an amplified boomerang distinguisher [16] for 36-round SHACAL-1. This distinguisher is used to attack 39-round SHACAL-1 using $2^{158.5}$ chosen plaintexts and about $2^{250.8}$ 39-round SHACAL-1 encryptions. The attack is based on guessing (or trying) the subkeys of the three additional rounds, and then checking whether the distinguisher succeeds. This approach is further extended to attack 47-round SHACAL-1 before exhaustive key search becomes faster than this attack. Another attack presented in [20] is a differential attack on 41-round SHACAL-1. The success of these attacks was questioned and resolved in [7].

Besides resolving the problems with previous attacks, in [7] a rectangle attack on 49-round SHACAL-1 is presented. The attack requires $2^{151.9}$ chosen plaintexts, and has a running time equivalent to $2^{508.5}$ 49-round SHACAL-1 encryptions.

In [18] a related-key rectangle attack with two keys is presented against 59-round SHACAL-1. This attack has a data complexity of $2^{149.7}$ related-key chosen plaintexts and has a time complexity of $2^{498.3}$ 59-round SHACAL-1 encryptions. This attack is improved in [15] to a related-key rectangle attack with four keys on 70-round SHACAL-1. The improved attack has a data complexity of $2^{151.8}$ related-key chosen plaintexts, and a time complexity of $2^{500.1}$ 70-round SHACAL-1 encryptions.

4 Related-Key Boomerang and Related-Key Rectangle Attacks

In this section we briefly describe the related-key rectangle attack. First, we outline the boomerang and the rectangle attacks and describe related-key differentials. Then, we describe the combination that forms into the related-key rectangle attack.

4.1 The Rectangle Attack

The rectangle attack [5] is an improved variant of the amplified boomerang attack [16] that has evolved from the boomerang attack presented in [27]. We first describe the boomerang attack, and then show the transformation into amplified boomerang/rectangle attacks.

The main idea behind the boomerang attack is to use two short differentials with high probabilities instead of one long differential with a low probability. We assume that a block cipher $E:\{0,1\}^n \times \{0,1\}^k \to \{0,1\}^n$ can be described as a cascade $E = E_1 \circ E_0$, such that for E_0 there exists a differential $\alpha \to \beta$ with probability p, and for E_1 there exists a differential $\gamma \to \delta$ with probability q.

The distinguisher is based on the following boomerang process:

- Ask for the encryption of a pair of plaintexts (P_1, P_2) such that $P_1 \oplus P_2 = \alpha$ and denote the corresponding ciphertexts by (C_1, C_2).
- Calculate $C_3 = C_1 \oplus \delta$ and $C_4 = C_2 \oplus \delta$, and ask for the decryption of the pair (C_3, C_4). Denote the corresponding plaintexts by (P_3, P_4).
- Check whether $P_3 \oplus P_4 = \alpha$.

The boomerang attack uses the first characteristic $(\alpha \to \beta)$ for E_0 with respect to the pairs (P_1, P_2) and (P_3, P_4), and uses the second characteristic $(\gamma \to \delta)$ for E_1 with respect to the pairs (C_1, C_3) and (C_2, C_4).

For a random permutation the probability that the last condition is satisfied is 2^{-n}. For E, the probability that the pair (P_1, P_2) is a right pair with respect to the first differential $(\alpha \to \beta)$ is p. The probability that both pairs (C_1, C_3) and (C_2, C_4) are right pairs with respect to the second differential is q^2. If all these are right pairs, then $E_1^{-1}(C_3) \oplus E_1^{-1}(C_4) = \beta = E_0(P_3) \oplus E_0(P_4)$. Thus, with probability p, $P_3 \oplus P_4 = \alpha$. The total probability of this quartet of plaintexts and ciphertexts to satisfy the boomerang conditions is $(pq)^2$.

The attack can be mounted for all possible β's and γ's simultaneously (as long as $\beta \neq \gamma$). Therefore, a right quartet for E is encountered with probability no less than $(\hat{p}\hat{q})^2$, where:

$$\hat{p} = \sqrt{\sum_{\beta} \Pr{}^2[\alpha \to \beta]}, \qquad \text{and} \qquad \hat{q} = \sqrt{\sum_{\gamma} \Pr{}^2[\gamma \to \delta]}.$$

The complete analysis is given in [27, 5, 6].

As the boomerang attack requires adaptive chosen plaintexts and ciphertexts, many of the techniques that were developed for using distinguishers in key recovery attacks can not be combined with the boomerang attack. This led to the introduction of chosen plaintext variants of the boomerang attack called the *amplified boomerang attack* [16] and the *rectangle attack* [5]. The transformation of the boomerang attack into a chosen plaintext attack is quite standard, and is achieved by birthday-paradox arguments. The key idea behind the transformation is to encrypt many plaintext pairs with input difference α, and to look for quartets that conform to the requirements of the boomerang process.

The rectangle (or the amplified boomerang) process is as follows:

- Ask for the encryption of many pairs of plaintexts $(P, P \oplus \alpha)$.
- Search two pairs of plaintexts $(P_1, P_2), (P_3, P_4)$, and their corresponding ciphertexts (C_1, C_2) and (C_3, C_4), respectively, satisfying:
  - $P_1 \oplus P_2 = P_3 \oplus P_4 = \alpha$
  - $C_1 \oplus C_3 = C_2 \oplus C_4 = \delta$

Given the same decomposition of E as before, and the same basic differentials, the analysis in [5] shows that out of N plaintext pairs, the number of right quartets is expected to be $N^2 2^{-n} \hat{p}^2 \hat{q}^2$. We note, that the main reduction in the probability follows from the fact that unlike the boomerang attack, in the rectangle attack the event $E_0(P_1) \oplus E_0(P_3) = \gamma$ occurs with probability 2^{-n} even when all the differentials hold.

4.2 Related-Key Differentials

Related-key differentials [17] were used for cryptanalysis several times in the past. Recall, that a regular differential deals with some plaintext difference ΔP and a ciphertext difference ΔC such that

$$\Pr{}_{P,K}[E_K(P) \oplus E_K(P \oplus \Delta P) = \Delta C]$$

is high enough (or zero [2]).

A related-key differential is a triplet of a plaintext difference ΔP, a ciphertext difference ΔC, and a key difference ΔK, such that

$$\Pr{}_{P,K}[E_K(P) \oplus E_{K \oplus \Delta K}(P \oplus \Delta P) = \Delta C]$$

is useful (high enough or zero).

4.3 Related-Key Rectangle Attack

The related-key rectangle attack was introduced in [18, 15], and independently in [8].

Let us assume that we have a related-key differential $\alpha \to \beta$ of E_0 under a key difference ΔK_{ab} with probability p. Assume also that we have another related-key differential $\gamma \to \delta$ for E_1 under a key difference ΔK_{ac} with probability q.

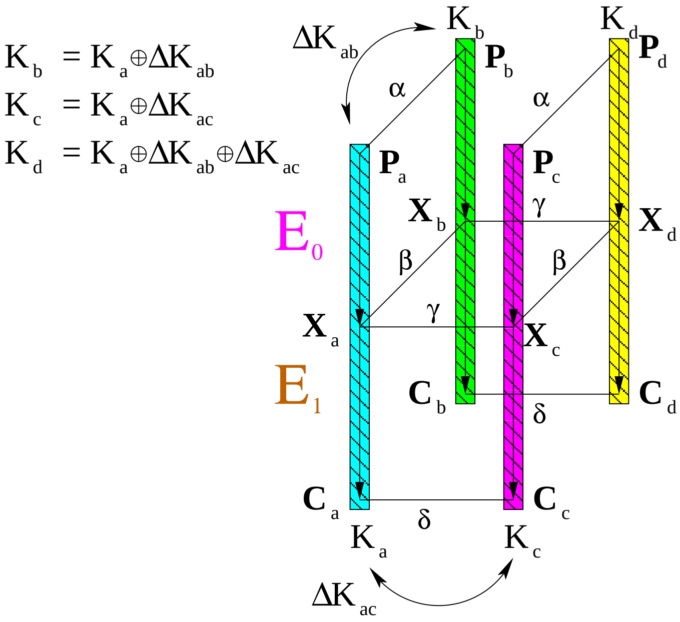

Fig. 1. A Related-Key Rectangle Quartet

The related-key rectangle process involves four different unknown (but related) keys — K_a, $K_b = K_a \oplus \Delta K_{ab}$, $K_c = K_a \oplus \Delta K_{ac}$, and $K_d = K_a \oplus \Delta K_{ab} \oplus \Delta K_{ac}$. The attack is performed by the following algorithm:

- Choose N plaintext pairs $(P_a, P_b = P_a \oplus \alpha)$ at random and ask for the encryption of P_a under K_a and of P_b under K_b. Denote the set of these pairs by S.
- Choose N plaintext pairs $(P_c, P_d = P_c \oplus \alpha)$ at random and ask for the encryption of P_c under K_c and P_d under K_d. Denote the set of these pairs by T.
- Search a pair of plaintexts $(P_a, P_b) \in S$ and a pair of plaintexts $(P_c, P_d) \in T$, and their corresponding ciphertexts (C_a, C_b) and (C_c, C_d), respectively, satisfying:
  - $P_a \oplus P_b = P_c \oplus P_d = \alpha$
  - $C_a \oplus C_c = C_b \oplus C_d = \delta$

See Figure 1 for an outline of such a quartet.

The attack can use many differentials for E_0 and E_1 simultaneously (just like in a regular rectangle attack) as long as all related-key differentials used in E_0 have the same key difference ΔK_{ab} and the same input difference α and as long as all related-key differentials used in E_1 have the same key difference ΔK_{ac} and the same output difference δ.

The analysis of the related-key rectangle attack is similar to the one of the rectangle attack. Starting with N plaintext pairs in S and N plaintext pairs in T, we expect to find $N^2 2^{-n}(\hat{p}\hat{q})^2$ right quartets in $S \times T$. For a random permutation the number of "right quartets" is about $N^2 2^{-2n}$, so as long as $\hat{p}\hat{q} > 2^{-n/2}$ we can use the related-key rectangle attack to distinguish between a random permutation and the attacked cipher. This distinguisher can be later used for a key recovery attack.

We note that a related-key boomerang attack can be constructed similarly to the related-key rectangle attack. The full analysis can be found in [8]. The related-key boomerang and rectangle techniques were used to attack reduced round variants of AES, IDEA, SHACAL-1, and SHACAL-2 and the full KASUMI and COCONUT98 [8, 9, 15, 19, 18].

In the case of SHACAL-1, the key schedule algorithm is linear. Therefore, given a key difference, all subkey differences are known, and can be easily used in the related-key model.

5 Related-Key Rectangle Attack on the Full SHACAL-1

Our attack on SHACAL-1 is based on a 69-round related-key distinguisher. In the attack on the full SHACAL-1, we try all the possible subkeys of the remaining 11 rounds, and decrypt all the ciphertexts. Then, the 69-round distinguisher is applied. We improve the time complexity of the attack by partially decrypting only 8 rounds, and then use the early abort approach to reduce the number of values that are decrypted through the remaining three more rounds, before the attack is applied. It is expected that for the right guess of the subkey of the last 11 rounds, the distinguisher would be more successful than for a wrong guess. Thus, we can use this distinguisher to identify (to some extent) the right subkey.

5.1 69-Round Related-Key Distinguisher

We decompose 69-round SHACAL-1 into two sub-ciphers: E_0 that contains the first 34 rounds of SHACAL-1 (rounds 0–33), and E_1 that contains the remaining 35 rounds (rounds 34–68).

We have transformed the collision producing differentials of SHA-1 presented in [30] into related-key differentials for each of the two sub-ciphers. The first related-key differential (for E_0) has probability 2^{-41}, and by fixing two bits of the plaintexts and using several differentials simultaneously for E_0 we obtain $\hat{p} = 2^{-38.5}$. The second related-key differential (for E_1) has probability 2^{-39}, and by using several differentials simultaneously for E_1 we obtain $\hat{q} = 2^{-38.3}$. The differentials are presented in Appendix A.

Combining these two differentials together leads to a 69-round related-key rectangle distinguisher with probability $2^{-80} \cdot \hat{p}\hat{q} = 2^{-156.8}$, i.e., given N related-key chosen plaintext pairs, we expect $N^2 \cdot 2^{-160} \cdot (\hat{p}\hat{q})^2$ right quartets. Hence, given $2^{157.8}$ related-key chosen plaintext pairs, we expect four right rectangle quartets, while for a random cipher only $2^{-4.4}$ are expected.

5.2 The Key Recovery Attack

The basic approach for a key recovery attack is to guess the subkey of the last 11 rounds, partially decrypt all ciphertexts, and apply the distinguisher for the remaining 69 rounds. Such an approach can be improved using the fact that in every round, only a small part of the intermediate value is substantially changed, while most of the value is only shifted. The attack is based on the *early abort* technique which is widely used [11, 12]. In this technique, once a pair/quartet does not satisfy the required differences/properties it is excluded from further analysis.

In the description of the attack algorithm we use the following notations: X_A denotes the value of word A in X. Similarly, $Y_{D,E}$ denotes words D and E of Y, etc. Let ΔX_i denote the difference in word X before round i, i.e., ΔA_{70} is the difference in word A before round 70, and after round 69. Also, let e_i be the 32-bit word composed of 31 0's and 1 in the ith place. We use $e_{i,j}$ to denote $e_i \oplus e_j$ and $e_{i,j,k} = e_{i,j} \oplus e_k$, etc. We also denote the set of possible values of ΔA_{70} given that the second differential is satisfied by S'.

We observe that even if we partially decrypt only 8 rounds, we still have a filtering condition on the quartets: Since $\Delta D_{72} = ROTL_{30}(\Delta A_{69})$ and $\Delta E_{72} = ROTL_{30}(\Delta B_{69})$, we can check whether the difference in these words corresponds to the output difference in words A and B of the second differential. In addition, we observe that we can extend the second differential by a truncated differential of one additional round. There are only $324 = 2^{8.3}$ possible ΔA_{70} values in S', hence, there are only 324 possible values for ΔC_{72} in case the second differential holds.

Using these observations, we can get a filtering of $64 + 23.7 = 87.7$ bits for every pair in the end of round 71, or a filtering of 175.4 bits in total. Since the attack starts with $2^{315.6}$ quartets, we expect that $2^{140.3}$ quartets pass the filtering for any given subkey guess of rounds 72–79. We then guess the subkey of round 71 and compute ΔE_{71} that is equal to ΔC_{69} if the differential holds to obtain an additional 64-bit filtering on the remaining quartets. After this filtering only $2^{76.3}$ quartets remain for each subkey guess. Then we continue by guessing the subkeys of rounds 70 and 69. As a result, the time complexity of the attack drops rapidly, while the data complexity remains unchanged.

The algorithm of the attack is as follows:

1. **Data Collection Phase:**
   (a) Ask for the encryption of $2^{157.8}$ pairs of plaintexts (P_a, P_b), where $P_b = P_a \oplus \alpha$, where P_a and P_b satisfy the restrictions described in Appendix A, and where P_a is encrypted under K_a and P_b is encrypted under K_b.
   (b) Ask for the encryption of $2^{157.8}$ pairs of plaintexts (P_c, P_d), where $P_c = P_d \oplus \alpha$, where P_c and P_d satisfy the restrictions described in Appendix A, and where P_c is encrypted under K_c and P_d is encrypted under K_d.
2. **Partial Decryption:**
   (a) For each guess of the subkey of rounds 72–79:
      i. Partially decrypt all ciphertexts (under the corresponding keys).

ii. Find all pairs of partially decrypted ciphertexts (C_a, C_c), such that $C_{a_{C,D,E}} \oplus C_{c_{C,D,E}} \in S$, where C_a is encrypted under K_a, C_c is encrypted under K_c and $S = \{(x, y, z) : ROTL_{30}(x) \in S', ROTL_{30}(y) = \delta_A = 0, ROTL_{30}(z) = \delta_B = e_2\}$.
iii. For each such pair (C_a, C_c), let P_a and P_c be the corresponding plaintexts. Let $P_b = P_a \oplus \alpha$ and $P_d = P_c \oplus \alpha$, and let C_b and C_d be the corresponding ciphertexts, respectively.
iv. If $C_{b_{C,D,E}} \oplus C_{d_{C,D,E}} \in S$ pass the quartet (P_a, P_b, P_c, P_d) for a further analysis.

(b) **Partial Decryption of Round 71:** For each guess of the subkey of round 71:
  i. Partially decrypt all the remaining quartets (under the corresponding keys) and denote the resulting intermediate values by (C'_a, C'_b, C'_c, C'_d).
  ii. For each of the remaining quartets, check whether $C'_{a_E} \oplus C'_{c_E} = \delta_C = 0$ and discard all the quartets that do not satisfy the equation.
  iii. For each of the remaining quartets, check whether $C'_{b_E} \oplus C'_{d_E} = \delta_C = 0$ and discard all the quartets that do not satisfy the equation.

(c) **Partial Decryption of Round 70:** For each guess of the subkey of round 70:
  i. Partially decrypt all the remaining quartets (under the corresponding keys) and denote the resulting intermediate values by $(C''_a, C''_b, C''_c, C''_d)$.
  ii. For each of the remaining quartets, check whether $C''_{a_E} \oplus C''_{c_E} = \delta_D = 0$ and discard all the quartets that do not satisfy the equation.
  iii. For each of the remaining quartets, check whether $C'_{b_E} \oplus C'_{d_E} = \delta_D = 0$ and discard all the quartets that do not satisfy the equation.

(d) **Partial Decryption of Round 69:** For each guess of the subkey of round 69:
  i. Partially decrypt all the remaining quartets (under the corresponding keys) and denote the resulting intermediate values by $(C'''_a, C'''_b, C'''_c, C'''_d)$.
  ii. For each of the remaining quartets, check whether $C'''_{a_E} \oplus C'''_{c_E} = \delta_E = e_1$ and discard all the quartets that do not satisfy the equation.
  iii. For each of the remaining quartets, check whether $C'''_{b_E} \oplus C'''_{d_E} = \delta_E = e_1$ and discard all the quartets that do not satisfy the equation.
  iv. Pass all the remaining quartets to further analysis.

(e) **Further Analysis:** If for this subkey guess only one quartet is suggested (or no quartets are suggested) discard the subkey guess. If the subkey is not discarded, exhaustively search all possible values for the remaining 160 subkey bits for the correct key.

5.3 Analysis of the Key Recovery Attack

The time complexity of Step 1 is $2^{159.8}$ encryptions. The time complexity of Step 2(a) is $\frac{8}{80} \cdot 2^{256} \cdot 2^{158.8} = 2^{411.5}$ SHACAL-1 encryptions. Steps 2(b)–2(e) are repeated for each subkey guess, i.e., 2^{256} times. For a given subkey guess, Step 2(b) consists of $2^{141.3} \cdot 2^{32}$ partial decryptions of one SHACAL-1 round. This is equivalent to $2^{141.3} \cdot 2^{32} \cdot \frac{1}{80} = 2^{167.0}$ full SHACAL-1 encryptions. Thus, the total time complexity of Step 2(b) is about $2^{256} \cdot 2^{167.0} = 2^{423.0}$ SHACAL-1 encryptions.

The time complexities of the other steps are relatively smaller. Hence, the total data complexity of the attack is $2^{159.8}$ related-key chosen plaintexts encrypted under four keys, and the time complexity is $2^{423.0}$ SHACAL-1 encryptions. The memory requirement of the attack is about $2^{159.8}$ memory blocks of 320 bits, required for storing the large amount of data.

We note that a different approach may be used in our attack. We can remove the last three rounds of the second differential to increase its probability by a factor of 2^6, resulting in a 66-round related-key rectangle distinguisher with probability $2^{-80} \cdot \hat{p} \cdot \hat{q} = 2^{-150.8}$. The resultant distinguisher requires $2^{151.8}$ related-key chosen plaintext pairs (P_a, P_b) and (P_c, P_d) each to produce four right plaintext quartets (while for a random cipher about $2^{-16.4}$ quartets that satisfy the rectangle conditions are expected). Then, we apply partial decryptions of rounds 69-79, 68, 67 and 66 in Steps 2(a), 2(b), 2(c) and 2(d), respectively, and then run the final exhaustive search for the remaining 64-bit keys in Step 2(e).

The time complexity of Step 2(a) in this case is $2^{152.8+352} \cdot (11/80) = 2^{501.9}$ SHACAL-1 encryptions. In this attack we can derive the set S in Step 2-(a) for the filtering of quartets, which has $2^{70.8}$ elements, and thus the number of remaining quartets after this step is about $(2^{151.9} \cdot 2^{-160+70.8})^2 = 2^{125.2}$. It follows that Step 2(b) takes about $2^{126.2} \cdot 2^{352+32} \cdot (1/80) = 2^{503.9}$ SHACAL-1 encryptions on average. Compared to Steps 2(a) and 2(b), the followed steps have quite small time complexities. Hence, this full-round attack on SHACAL-1 works with a data complexity of $2^{153.8}$ related-key chosen plaintexts encrypted under four related keys and with a time complexity of $2^{501.9} + 2^{503.9} = 2^{504.2}$ SHACAL-1 encryptions.

6 Differences Between Attacking SHA-1 and SHACAL-1

While it may seem that any attack on SHA-1 can be easily transformed into an attack on SHACAL-1, and vice versa, this is not exactly the case. Investigating the recent attacks on SHA-1 in [3, 4, 30], it seems that these attacks heavily rely on the fact that the attacker can control some of the bits that enter the nonlinear operations. This way, the collision-producing differentials have much higher probability than the respective related-key differentials we use. We can impose conditions on the keys (increasing the probabilities of the related-key differentials), but then our attack would be applicable only for such keys, i.e., a weak key class.

Another difference between the attacks on SHA-1 and our attack is the fact that the collision attacks can iteratively fix the values they use, i.e., using message modification techniques or neutral bits. This enables the collision producing attacks to use shorter differential than ours (as these attacks actually start the probabilistic process in a much later step).

There is another difference between the two cases. While in the case of encryption (SHACAL-1), we have to deal with each block of message independently, collision attacks on the hash function can use multiple blocks. For example, the attacker can treat messages that detoured the differential in a very late step, by respective changes to the second block of the message. This fact allows the collision search to use shorter differentials (this time from the end point), thus, increasing the success probability.

Another problem our attack faces is the dual representation of XOR and additive differentials. As we have less control on the encryption process than the collision attacks have on the compression process, it is less useful for us to consider the differentials using the dual representation. Again, for exploiting the advantage of the additive differentials in the related-key differentials, we must fix some of the key bits, resulting again in a weak key class.

7 Summary and Conclusions

In this paper we converted the differentials of the compression function of SHA-1 presented by Wang et al. to related-key differentials of the block cipher SHACAL-1. Then we used the related-key rectangle technique to devise the first known attack on the full 80-round SHACAL-1.

We also discussed the possibility of converting other techniques used in the attacks on SHA-1 to attack SHACAL-1, and concluded that such conversion will result in an attack applicable only to a weak key class of SHACAL-1.

Our attack improves by far the previously known results, that were able to attack up to 70 rounds of the cipher, and demonstrates the power of the related-key rectangle technique. However, the result is still highly theoretical and a practical attack on the full SHACAL-1 seems out of reach at this stage.

References

1. Eli Biham, *New Types of Cryptanalytic Attacks Using Related Keys*, Journal of Cryptology, Vol. 7, No. 4, pp. 229–246, Springer-Verlag, 1994.
2. Eli Biham, Alex Biryukov, Adi Shamir, *Cryptanalysis of Skipjack Reduced to 31 Rounds Using Impossible Differentials*, Advances in Cryptology, proceedings of EUROCRYPT '99, Lecture Notes in Computer Science 1592, pp. 12–23, Springer-Verlag, 1999.
3. Eli Biham, Rafi Chen, *Near-Collisions of SHA-0*, Advances in Cryptology, proceedings of CRYPTO 2004, Lecture Notes in Computer Science 3152, pp. 290–305, Springer-Verlag, 2004.
4. Eli Biham, Rafi Chen, Antoine Joux, Patrick Carribault, Christophe Lemuet, William Jalby, *Collisions of SHA-0 and Reduced SHA-1*, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3621, pp. 36–57, 2005.
5. Eli Biham, Orr Dunkelman, Nathan Keller, *The Rectangle Attack – Rectangling the Serpent*, Advances in Cryptology, proceedings of EUROCRYPT '01, Lecture Notes in Computer Science 2045, pp. 340–357, Springer-Verlag, 2001.
6. Eli Biham, Orr Dunkelman, Nathan Keller, *New Results on Boomerang and Rectangle Attacks*, proceedings of Fast Software Encryption 9, Lecture Notes in Computer Science 2365, pp. 1–16, Springer-Verlag, 2002.

7. Eli Biham, Orr Dunkelman, Nathan Keller, *Rectangle Attacks on 49-Round SHACAL-1*, proceedings of Fast Software Encryption 10, Lecture Notes in Computer Science 2887, pp. 22–35, Springer-Verlag, 2003.
8. Eli Biham, Orr Dunkelman, Nathan Keller, *Related-Key Boomerang and Rectangle Attacks*, Advances in Cryptology, proceedings of EUROCRYPT '05, Lecture Notes in Computer Science 3494, pp. 507–525, Springer-Verlag, 2005.
9. Eli Biham, Orr Dunkelman, Nathan Keller, *A Related-Key Rectangle Attack on the Full KASUMI*, Advances in Cryptology, proceedings of ASIACRYPT 2005, Lecture Notes in Computer Science 3788, pp. 443–461, 2005.
10. Eli Biham, Adi Shamir, *Differential Cryptanalysis of the Data Encryption Standard*, Springer-Verlag, 1993.
11. Florent Chabaud, Antoine Joux, *Differential Collisions in SHA-0*, Advances in Cryptology, proceedings of CRYPTO 1998, Lecture Notes in Computer Science 1462, pp. 56–71, Springer-Verlag, 1998.
12. Niels Ferguson, John Kelsey, Stefan Lucks, Bruce Schneier, Mike Stay, David Wagner, Doug Whiting, *Improved Cryptanalysis of Rijndael*, proceedings of Fast Software Encryption 7, Lecture Notes in Computer Science 1978, pp. 213–230, Springer-Verlag, 2001.
13. Helena Handschuh, Lars R. Knudsen, Matthew J. Robshaw, *Analysis of SHA-1 in Encryption Mode*, proceedings of CT-RSA 2001, Lecture Notes in Computer Science 2020, pp. 70–83, Springer-Verlag, 2001.
14. Helena Handschuh, David Naccache, *SHACAL*, preproceedings of NESSIE first workshop, Leuven, 2000.
15. Seokhie Hong, Jongsung Kim, Guil Kim, Sangjin Lee, Bart Preneel, *Related-Key Rectangle Attacks on Reduced Versions of SHACAL-1 and AES-192*, proceedings of Fast Software Encryption 12, Lecture Notes in Computer Science 3557, pp. 368–383, Springer-Verlag, 2005.
16. John Kelsey, Tadayoshi Kohno, Bruce Schneier, *Amplified Boomerang Attacks Against Reduced-Round MARS and Serpent*, proceedings of Fast Software Encryption 7, Lecture Notes in Computer Science 1978, pp. 75–93, Springer-Verlag, 2000.
17. John Kelsey, Bruce Schneier, David Wagner, *Key-Schedule Cryptoanalysis of IDEA, G-DES, GOST, SAFER, and Triple-DES*, Advances in Cryptology, proceedings of CRYPTO '96, Lecture Notes in Computer Science 1109, pp. 237–251, Springer-Verlag, 1996.
18. Jongsung Kim, Guil Kim, Seokhie Hong, Sangjin Lee, Dowon Hong, *The Related-Key Rectangle Attack — Application to SHACAL-1*, proceedings of ACISP 2004, Lecture Notes in Computer Science 3108, pp. 123–136, Springer-Verlag, 2004.
19. Jongsung Kim, Guil Kim, Sangjin. Lee, Jongin Lim and Junghwan Song, *Related-Key Attacks on Reduced Rounds of SHACAL-2*, Proceedings of INDOCRYPT 2004, Lecture Notes in Computer Science 3348, pp. 175–189, Springer-Verlag, 2004.
20. Jongsung Kim, Dukjae Moon, Wonil Lee, Seokhie Hong, Sangjin Lee, Seokwon Jung, *Amplified Boomerang Attack against Reduced-Round SHACAL*, Advances in Cryptology, proceedings of ASIACRYPT 2002, Lecture Notes in Computer Science 2501, pp. 243-253, Springer-Verlag, 2002.
21. NESSIE – New European Schemes for Signatures, Integrity and Encryption. *http://www.nessie.eu.org/nessie*
22. NESSIE, *Portfolio of recommended cryptographic primitives.*
23. NESSIE, *Performance of Optimized Implementations of the NESSIE Primitives*, NES/DOC/TEC/WP6/D21/2.

24. Markku-Juhani O. Saarinen, *Cryptanalysis of Block Ciphers Based on SHA-1 and MD5*, proceedings of Fast Software Encryption 10, Lecture Notes in Computer Science 2887, pp. 36–44, Springer-Verlag, 2003.
25. US National Bureau of Standards, *Secure Hash Standard*, Federal Information Processing Standards Publications No. 180-2, 2002.
26. Eteinee Van Den Bogeart, Vincent Rijmen, *Differential Analysis of SHACAL*, NESSIE internal report NES/DOC/KUL/WP3/009/a, 2001.
27. David Wagner, *The Boomerang Attack*, proceedings of Fast Software Encryption 6, Lecture Notes in Computer Science 1636, pp. 156–170, 1999.
28. Xiaoyun Wang, Andrew C. Yao, Frances Yao, *Cryptanalysis on SHA-1*, Cryptographic Hash Workshop, NIST, Gaithersburg, 2005.
29. Xiaoyun Wang, Xuejia Lai, Dengguo Feng, Hui Chen, Xiuyuan Yu, *Cryptanalysis of the Hash Functions MD4 and RIPEMD*, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3494, pp. 1–18, 2005.
30. Xiaoyun Wang, Yiqun Lisa Yin, Hongbo Yu, *Finding Collisions in the Full SHA-1*, Advances in Cryptology, proceedings of CRYPTO 2005, Lecture Notes in Computer Science 3621, pp. 17–36, 2005.
31. Xiaoyun Wang, Hongbo Yu, *How to Break MD5 and Other Hash Functions*, Advances in Cryptology, proceedings of EUROCRYPT 2005, Lecture Notes in Computer Science 3494, pp. 19–35, 2005.
32. Xiaoyun Wang, Hongbo Yu, Yiqun Lisa Yin, *Efficient Collision Search Attacks on SHA-0*, Advances in Cryptology, proceedings of CRYPTO 2005, Lecture Notes in Computer Science 3621, pp. 1–16, 2005.

A Related-Key Differentials of SHACAL-1

In this appendix we describe the differentials used for our related-key rectangle attacks on SHACAL-1. These differentials are based on the collision producing differentials presented in [30].

The first related-key differential is for rounds 0–33 and is presented in Table 2. The probability of the differential is 2^{-41}. This probability can be increased by a factor of 4 by fixing the equivalent to two bits in each of the plaintexts of the pair. If we set the most significant bit of A to be zero, the probability of the second round of the differential is increased by a factor of 2. By setting bit 3 of A to differ from bit 3 of B, the probability of the third round of the differential is also increased by a factor of 2.

We use the notation e_i to represent the 32-bit word composed of 31 0's and 1 in the ith place. We use $e_{i,j}$ to denote $e_i \oplus e_j$ and $e_{i,j,k} = e_{i,j} \oplus e_k$, etc.

Due to the nature of the rectangle attack, we can improve the probability by counting over several differentials. We have counted over differentials which have the same first 33 rounds as the differential presented in Table 2. The resulting probability is $\hat{p} = 2^{-38.5}$ (when fixing the respective bits of the plaintext).

The second related-key differential for rounds 34–68 is presented in Table 3. This differential is also based on the collision producing differentials of [30]. The probability of this differential is 2^{-39}.

Again, due to the nature of the rectangle attack, we can improve the probability by counting over several differentials. We count over various similar charac-

Round (i)	ΔK	ΔA_i	ΔB_i	ΔC_i	ΔD_i	ΔE_i	Probability
Input 0	e_1	0	0	e_{31}	e_{31}	e_{31}	2^{-1}
$1^\dagger$	e_6	e_1	0	0	e_{31}	e_{31}	2^{-2}
$2^\dagger$	$e_{1,31}$	0	e_1	0	0	e_{31}	2^{-2}
3	e_{31}	0	0	e_{31}	0	0	2^{-1}
4	$e_{1,31}$	0	0	0	e_{31}	0	2^{-2}
5	$e_{6,31}$	e_1	0	0	0	e_{31}	2^{-1}
6	0	0	e_1	0	0	0	2^{-2}
7	$e_{6,31}$	e_1	0	e_{31}	0	0	2^{-2}
8	e_{31}	0	e_1	0	e_{31}	0	2^{-3}
9	e_6	e_1	0	e_{31}	0	e_{31}	2^{-2}
10	e_{31}	0	e_1	0	e_{31}	0	2^{-3}
11	e_6	e_1	0	e_{31}	0	e_{31}	2^{-2}
12	$e_{1,31}$	0	e_1	0	e_{31}	0	2^{-3}
13	0	0	0	e_{31}	0	e_{31}	2^{-1}
14	e_{31}	0	0	0	e_{31}	0	2^{-1}
15	e_{31}	0	0	0	0	e_{31}	1
16	0	0	0	0	0	0	1
17	0	0	0	0	0	0	1
18	0	0	0	0	0	0	1
19	0	0	0	0	0	0	1
20	0	0	0	0	0	0	1
21	0	0	0	0	0	0	1
22	0	0	0	0	0	0	1
23	0	0	0	0	0	0	1
24	0	0	0	0	0	0	1
25	0	0	0	0	0	0	1
26	e_2	0	0	0	0	0	2^{-1}
27	e_7	e_2	0	0	0	0	2^{-1}
28	e_2	0	e_2	0	0	0	2^{-1}
29	$e_{0,3}$	0	0	e_0	0	0	2^{-2}
30	$e_{0,8}$	e_3	0	0	e_0	0	2^{-2}
31	$e_{0,3}$	0	e_3	0	0	e_0	2^{-2}
32	$e_{1,4}$	0	0	e_1	0	0	2^{-2}
33	$e_{1,9}$	e_4	0	0	e_1	0	2^{-2}
Output (34)		0	e_4	0	0	e_1	

† — The probability of this round can be improved by a factor of 2.
Differences are presented before the round, i.e., ΔA_0 is the input difference.

Table 2. Related-Key Differential for Rounds 0–33 of SHACAL-1.

teristics, by changing the first round of this differential. The resulting probability is $\hat{q} = 2^{-38.3}$.

Round (i)	ΔK	ΔA_i	ΔB_i	ΔC_i	ΔD_i	ΔE_i	Probability
Input 34	$e_{1,30}$	0	e_1	e_{31}	0	$e_{30,31}$	2^{-2}
35	e_1	0	0	e_{31}	e_{31}	0	2^{-1}
36	e_6	e_1	0	0	e_{31}	e_{31}	2^{-1}
37	$e_{1,31}$	0	e_1	0	0	e_{31}	2^{-1}
38	e_{31}	0	0	e_{31}	0	0	1
39	$e_{1,31}$	0	0	0	e_{31}	0	2^{-1}
40	$e_{6,31}$	e_1	0	0	0	e_{31}	2^{-1}
41	0	0	e_1	0	0	0	2^{-2}
42	$e_{6,31}$	e_1	0	e_{31}	0	0	2^{-2}
43	e_{31}	0	e_1	0	e_{31}	0	2^{-3}
44	e_6	e_1	0	e_{31}	0	e_{31}	2^{-2}
45	e_{31}	0	e_1	0	e_{31}	0	2^{-3}
46	e_6	e_1	0	e_{31}	0	e_{31}	2^{-2}
47	$e_{1,31}$	0	e_1	0	e_{31}	0	2^{-3}
48	0	0	0	e_{31}	0	e_{31}	2^{-1}
49	e_{31}	0	0	0	e_{31}	0	2^{-1}
50	e_{31}	0	0	0	0	e_{31}	1
51	0	0	0	0	0	0	1
52	0	0	0	0	0	0	1
53	0	0	0	0	0	0	1
54	0	0	0	0	0	0	1
55	0	0	0	0	0	0	1
56	0	0	0	0	0	0	1
57	0	0	0	0	0	0	1
58	0	0	0	0	0	0	1
59	0	0	0	0	0	0	1
60	0	0	0	0	0	0	1
61	e_2	0	0	0	0	0	2^{-1}
62	e_7	e_2	0	0	0	0	2^{-1}
63	e_2	0	e_2	0	0	0	2^{-1}
64	$e_{0,3}$	0	0	e_0	0	0	2^{-2}
65	$e_{0,8}$	e_3	0	0	e_0	0	2^{-2}
66	$e_{0,3}$	0	e_3	0	0	e_0	2^{-2}
67	$e_{1,4}$	0	0	e_1	0	0	2^{-2}
68	$e_{1,9}$	e_4	0	0	e_1	0	2^{-2}
Output 69		0	e_4	0	0	e_1	

Differences are presented before the round, i.e., ΔA_{34} is the input difference.

Table 3. Related-Key Differential for Rounds 34–68 of SHACAL-1.